Self-tuning Fuzzy Task Space Controller for Puma 560 Robot

Azita Azarfar
Faculty member in Department of Electrical and Computer Engineering, Shahrood branch, Islamic Azad University, Iran

Article Info

Article history:

Received Sep 29, 2018
Revised Nov 15, 2018
Accepted Nov 29, 2018

Keyword:

Inverse jacobian method
Puma 560
Robot task space
Robot tracking
Self-tuning fuzzy system

ABSTRACT

Since in most robot applications the desired paths are determined in task space or Cartesian space, it is important to control the robot arm in task space. In this paper a fuzzy controller with modifiable scaling factors is proposed to control the robot end-effector in task space. The controller is a fuzzy system with a mechanism to change the scaling factors when the error is bounded under a predetermined value. The controller is designed in joint space and is developed to work space by using inverse Jacobian strategy. The simulations results on Puma 560 robot manipulator illustrate the high performance of presented control method.

Corresponding Author:

Azita Azarfar,
Faculty member in Department of Electrical and Computer Engineering,
Shahrood branch, Islamic Azad University, Iran.
Email: Azita.azarfar@gmail.com

1. INTRODUCTION

In most robot applications, a desired trajectory for robot end effecter is designed in task space or Cartesian space. However, the joint coordinates contain the control actions. To design a joint space controller for task objectives, it is necessary to carry out the inverse kinematics transformation to obtain the joint space desired path. To avoid solving the inverse kinematic problem some researchers used inverse or transpose Jacobian method to control a robot end-effector to a set point or to a trajectory in task space [1, 2]. In [3] an adaptive Jacobian controller is proposed with uncertainty in kinematics. Several research works on approximate Jacobian matrix in robot control have also are presented [4-6]. It should be noticed that an inexact Jacobian matrix results in imprecise transformation from task-space to joint-space and it creates some difficulties in control design.

The performance of control strategies has been improved by using intelligent control systems. They have the ability to overcome the limits of classical and adaptive methods. The model dependency is usually decreased by using these methods. Some of intelligent methods such as fuzzy systems have in-built adaption and decision making capabilities to overcome the uncertainty. As robotics problems are nonlinear, many research works developed intelligent methods for robot control. For example, [7] used the indirect adaptive fuzzy method to control the contact force. Fuzzy sliding mode control is used to control an unknown robot [8]. The authors in [9] proposed a sliding mode control system in which the coefficients of sliding surface are optimized by Particle Swarm algorithm, and the method is combined with neuro fuzzy system to guarantee the robustness. Type-2 fuzzy systems are also used by [10] for robot trajectory control.

In this paper, a fuzzy system with modifiable scaling factors is proposed to control the robot in task space. Moreover, the motor control is used to decentralize and control the robot. Three control systems are designed for motors which are attached to three robot joints. Whenever the tracking error is bounded under fixed values, the input scaling factor is increasing linearly to effectively decrease the error. The presented fuzzy system is a Takagi Sugeno type with fixed values in consequent parts except in zero fuzzy set.

The consequent part of the rule related to zero fuzzy sets is PD controller which operates in adaptive manner due to scaling factor mechanism. The objective of control system is that the end-effector of robot manipulator track a desired trajectory in task space. The Jacobian inverse method is used for transferring the Cartesian tracking error to joint angles error. Simulations have been performed to investigate the performance of the proposed control method. The control system is applied to a 3-DOF Puma 560 robotic arm in 2 steps. At the first step, the model is considered completely known and precise. In the second steps up to 50 % uncertainty is considered in robot parameters. In both simulations, the proposed control system has satisfactory performance in tracking a desired trajectory in task space.

The paper is organized as follows. Section 2 briefly introduced the Puma 560 robot manipulator. The problem formulation is discussed in Section 3. The control system is designed in Section 4. Section 5 presents the simulations outcomes, and Section 6 conclude the paper.

2. PUMA560 ROBOT

The PUMA 560 (Programmable Universal Manipulation Arm) is an industrial robot arm with six degree of freedom resulted from six rotational joints. It owns three joints for arm and three joints for wrist, as shown in Figure 1 [11]**.** Since the purpose of this paper is just to control the position of robot end point in task space, the robot arm joints are only considered in robot model. The Denavit-Hartenberg parameters of Puma robot are shown in Table 1 where for the considered robot in this paper, parameters are considered as $a_2 = 0.4318(\mathrm{m})$ ، $a_3 = 0.0203(\mathrm{m})$ ، $d_3 = 0.15005(\mathrm{m})$ and $d_4 = 0.43180(\mathrm{m})$. By using these parameters, the robot Jacobian matrix can be calculated. The Jacobian matrix represents the relation between angel velocities in joint space and Cartesian velocities in task space.

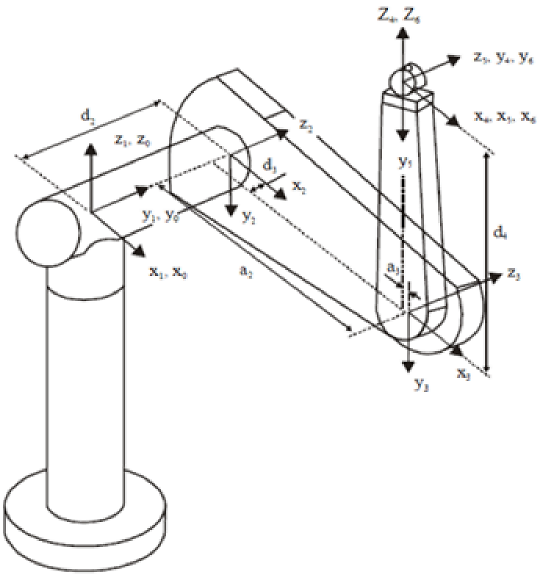

Figure 1. Puma 560 robot manipulator in zero position [11]

Table 1. Denavit-Hartenberg Parameters of Puma Robot

Joint i	θ	α_i	*a_i(m)*	*d_i(m*
1	θ_1	90	0	0
2	θ_2	0	a_2	0
3	θ_3	-90	a_3	d_3

3. PROBLEM FORMULATION

The simulated robot manipulator is a PUMA 560 robotic arm, which have three degrees of freedom (DOF) by three revolute joints. The robot manipulator is driven by DC motors, somehow each joint is run by one motor using a fuzzy controller. The method is called independent joint strategy and has been frequently used to apply fuzzy controllers [12]. The voltage of each motor is the output of each proposed controller for three joints. The differential equation of a DC motor is in the form of (1-2)

$$v = R_a I_a + L_a \dot{I}_a + K_b r^{-1} \dot{q}_i \tag{1}$$

$$I_a = K_m^{-1} T_m \tag{2}$$

where v is the motor voltage, I_a is the armature current, $\dot{q}_i$ is the velocity of *ith* joint. R_a,L_a are armature resistance and inductance, K_b, r are back-emf constant and reduction gear coefficient. K_{m} is the torque constant, and $\mathrm{T_m}$ is the motor torque to drive the *ith* joint of the robot manipulator stated as (3):

$$T_m = J_m \ddot{\theta}_m + B_m \dot{\theta}_m + rT_i \tag{3}$$

In (3), $\mathrm{T_i}$ is a member of $T\epsilon R^n$ that is the manipulator torque vector, $\mathrm{J_m}$ is the sum of rotor inertia and gear inertias, and $\mathrm{B_m}$ is the damping coefficient. The manipulator torque vector is given by(4) [13]

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{T} \tag{4}$$

where $\boldsymbol{q} \in R^n$ are the generalized coordinates, $\boldsymbol{D}(\boldsymbol{q}) \in R^{n\times n}$ is the symmetric positive-definite inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ represents the Coriolis and centripetal terms, and **g(q)** is a vector of gravitational torques.

In most robot applications a desired path for robot end-effector is specified in task space. Assume that $X \in R^n$ represents the task space vector of robot end-effector. Many commercial sensors are available for measurement of X, such as vision systems, electromagnetic measurement systems, position sensitive detectors or laser tracking systems. X is related to joint variables through robot kinematics equations. However the task space velocity $\dot{X}$ is related to the joint-space velocity $\dot{\boldsymbol{q}}$ as follows (5):

$$\dot{\boldsymbol{X}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{5}$$

where *J(q)* is the Jacobian matrix from joint space to task space. In small errors (5) can be rewritten as (6)

$$\Delta\boldsymbol{X} = \boldsymbol{J}\,(\boldsymbol{q})\,\Delta\boldsymbol{q} \tag{6}$$

where $\Delta\boldsymbol{X} = \boldsymbol{X_d} - \boldsymbol{X}$, $\Delta\boldsymbol{q} = \boldsymbol{q_d} - \boldsymbol{q}$, and $\boldsymbol{X_d}$ is the desired path in task space and $\boldsymbol{q_d}$ is the desired trajectory for joints corresponding to $\boldsymbol{X_d}$**.** So, $\Delta\boldsymbol{X}$ can be named as task space error and $\Delta\boldsymbol{q}$ as joint space error. Equation 6 yields (7)

$$\Delta\boldsymbol{q} = \boldsymbol{J^{-1}}\,(\boldsymbol{q})\,\Delta\boldsymbol{X} \tag{7}$$

where $\boldsymbol{J^{-1}}$ is the inverse of the Jacobian matrix. So, a relation between task space error and joint space errors can be calculated. This method is called Inverse Jacobian strategy. The robot arm has three joints and just the position control is aimed, so the Jacobian matrix would be 3 × 3. It is assumed that the robot is operating in a finite task space such that the Jacobian matrix is full rank. The main difficulty of estimating the joint space error from task space error comes from the fact that the Jacobian matrix is uncertain in the presence of kinematic uncertainties. So, the transformation will be imprecise in presence of uncertainties. The control objective for the Puma robot manipulator is hence to drive the manipulator endpoint to follow a desired trajectory defined in Cartesian space, in the presence of possibly large uncertainties in system kinematics and dynamics. The control approach is illustrated in Figure 2. The tracking error appears in task space, then by use of inverse Jacobian matrix, these errors are transferred to joint space and will be used as joint fuzzy adaptive controllers' inputs.

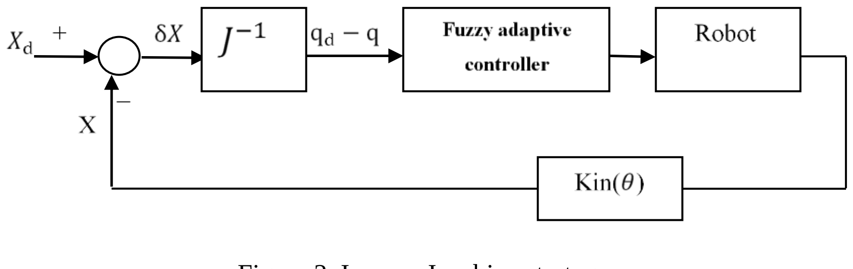

Figure 2. Inverse Jacobian strategy

4. CONTROL SYSTEM

The control system is composed of a TS controller and a mechanism for regulating scaling factors. When the tracking error is in the acceptable rang, the scaling factors are increased linearly to reduce the tracking error. On the other hand, if the tracking error is beyond the predetermined range for example due to initial conditions, the scaling factors are set with initial values.

Since this kind of gain rising results in sensitivity increasing of the system, the controller is designed in a way that whenever disturbance or any chaos increase the error of the system the scaling factors are reduced and back to the initial values. The TS fuzzy controller is designed with nine fuzzy rules in the form of (8):

$$R_j : if\ z_1\ is\ A_1^j\ and\ z_2\ is\ A_2^j\ THEN\ y = a_1^j z_1 + a_2^j z_2 + a_0^j \quad (8)$$

where R_j denotes the jth fuzzy rule, A_i^j is the fuzzy set, a_i^j is constant, y is the crisp output, for $j = 1,2,\dots,9$ and $i = 1,2$. The inputs are $z_1 = A_1(q_d - q)$ and $z_2 = A_2(\dot{q_d} - \dot{q})$ and $z = [z_1, z_2]^T$, while z are the transmitted errors from task space to joint space by inverse Jacobian matrix and A_i is the scaling factor for ith controller input that is adapted during the control process. The output $f(z) \in V \subset R$ of the TS fuzzy system is in form of (9)

$$\mathrm{f(z)} = \frac{\sum_{j=1}^{9}(a_1^j z_1 + a_2^j z_2 + a_0^j)(\prod_{i=1}^{n} \mu_{A_i^j}(z_i))}{\sum_{j=1}^{9}(\prod_{i=1}^{2} \mu_{A_i^j}(z_i))} \quad (9)$$

When the size of joint space error becomes less than a predetermined value, increasing the scaling factors is runs. If the tracking error is not reduced then the scaling factor mechanism will not run. Therefore, it is required that the TS controllers be stable. In [14], it is shown that the stability would be obtained if for $j = 1,2,\dots,9$:

a. $a_1^j \geq 0$

b. a_0^j should be as same sign as z_2

Through this condition, the TS fuzzy system with fixed scaling factor is stable. The proposed approach is illustrated in Figure 3. The TS fuzzy controller has two inputs: position error z_1 and its derivative z_2, and the motor voltage v as output. The nine fuzzy rules are given in Table 2. Membership functions (MFs) for the inputs are shown in Figure 4. The fuzzy system as a nonlinear function of mapping from input vector to the output is shown in Figure 5. The control system with the initial scaling factors must be stable. When error is decreased to a predetermined value in the zero fuzzy set, the scaling factor is increased linearly to improve the performance of the control system. As a result, the tracking error is decreased.

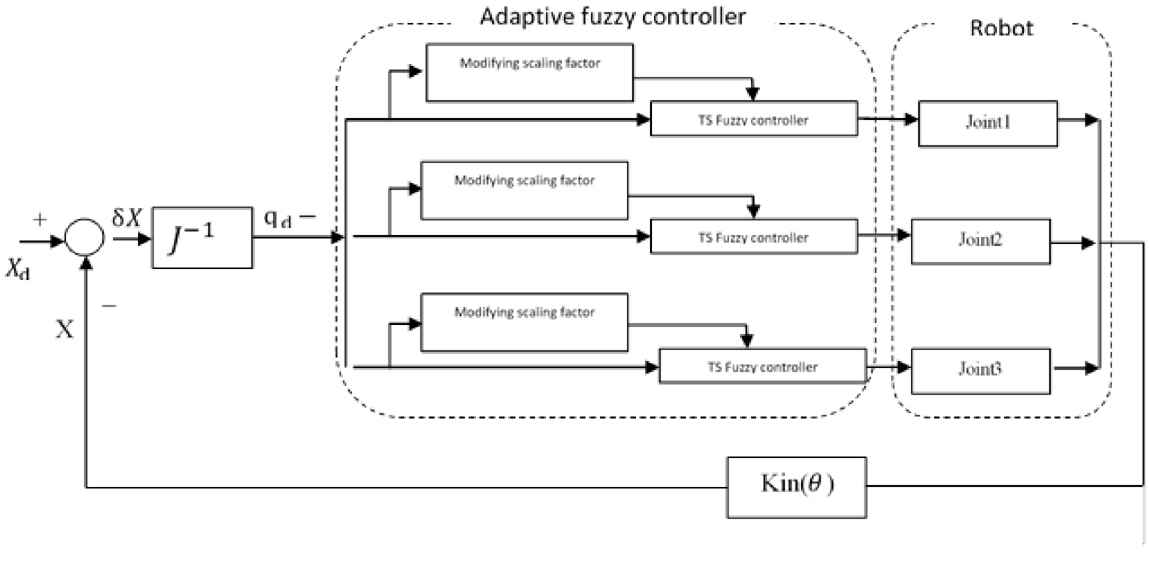

Figure 3. Control system

Table 2. Fuzzy Rules

Erro *Chang of error*	Positive	Zero	Negative
Positive	12	9	6
Zero	3	$9z_1 + z_2$	-3
Negative	-6	-9	-12

5. SIMULATION RESULTS

In this section, simulation results are provided to show the feasibility of the control scheme proposed in this paper. This controller is implemented on Puma560 robot arm. The control task is that the end-point of the robot manipulator is required to follow a circle trajectory defined in Cartesian space as (10):

$$\begin{cases} x = 0.203\sin(\pi t) \\ y = 0.203\cos(\pi t) \\ z = 0.4013 \end{cases} \tag{10}$$

The schematic model of Puma robot and the desired trajectory are shown in Figure 6. It totally takes two seconds for robot to complete the proposed trajectory. The robot model is simulated in Matlab, and the motor parameters of Puma robot are according to Table 3.

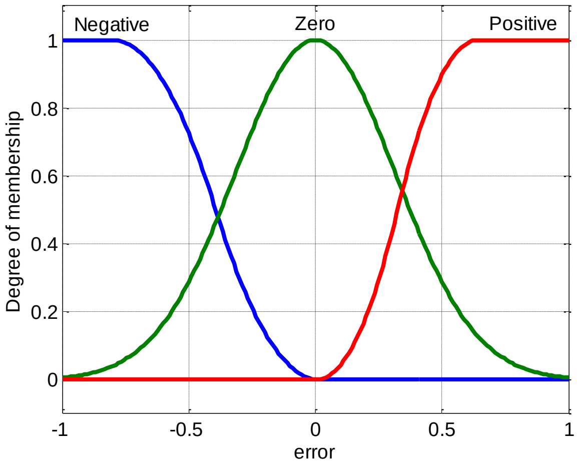

Figure 4. Membership functions for inputs

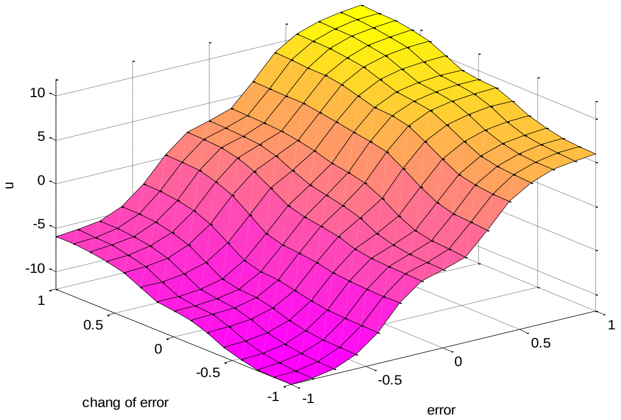

Figure 5. Fuzzy system rule surface

Table 3. Parameters for Joint Motors

Joint	R_a	L_a	J_m	B_m	K_m	r	K_b
1	1.6	0.0048	0.0002	0.00148	0.2611	62.55	0.19
2	1.6	0.0048	0.0002	0.000817	0.2611	107.81	0.19
3	1.6	0.0048	0.0002	0.00138	0.2611	53.15	0.19

The initial position of robot end point is considered in $[x_0, y_0, z_0] = [0.0097 \quad 0.15 \quad 0.42](m)$ in Cartesian space. Knowing this initial condition, the initial errors in task space due to desired trajectory are $[e_x, e_y, e_z] = [-0.00975 \quad 0.05242 \quad -0.02232](m)$. The three controllers are the same. The initial scaling factors for the input is set equal to 10 and for the output equal to 5. The threshold for absolute value of joint space error to run the scaling factor mechanisms for joints 1, 2, 3 are equal to 7.9858e-004rad, 3.7101e-004rad and 9.3099e-004rad, respectively. It should be noticed that uncertainty in the cinematic parameters makes the Jacobian matrix inexact, so the transformation would be imperfect, therefore, the accurate error will not be transmitted to joint space. To investigate the proposed control method performance more deeply, two simulations have been done. In first simulation all the cinematic parameters needed for calculating Jacobian matrix are considered to be exact. In the second one, up to 50 % uncertainty is considered for robot parameters and an external disturbance is also added to system.

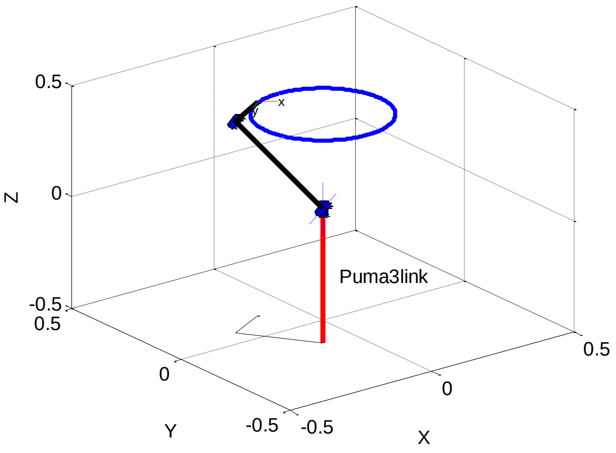

Figure 6. Puma Robot schematic model and the desired trajectory

5.1. Simulation A

In this part it is assumed that the cinematic parameters are certain and the Jacobian matrix is exact. The control objective is to track the circle in task-space. The performance of the control system is satisfactory as shown in Figure 7. The tracking errors are displayed in Figure 8. The steady state error is less than 0.2 millimeter. The operation of scaling factors based on joint space errors are illustrated in Figure 9- Figure 11 for different joints. The added arrows in the figures show the time of start of increasing mechanism. The input scaling factors increase linearly when the error is decreased enough. The Robot behaves very well with an ignorable tracking error due to the use of increasing scaling factors.

5.2. Simulation B

To show the effectiveness of the proposed controller against the uncertainty and external disturbance a simulation is run assuming that kinematic parameters are inexact. So the Jacobian matrix is not accurate and wrong error is transmitted to joint space.

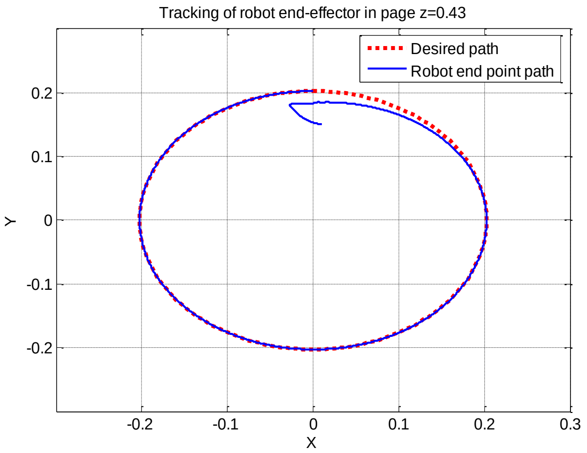

Figure 7. Performance of control system in simulation A, the robot end-effector successfully track the desired trajectory

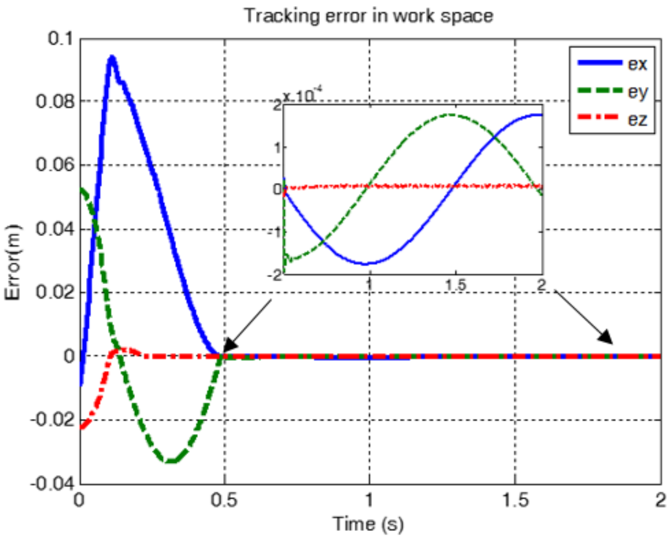

Figure 8. Tracking error in task space in simulation A

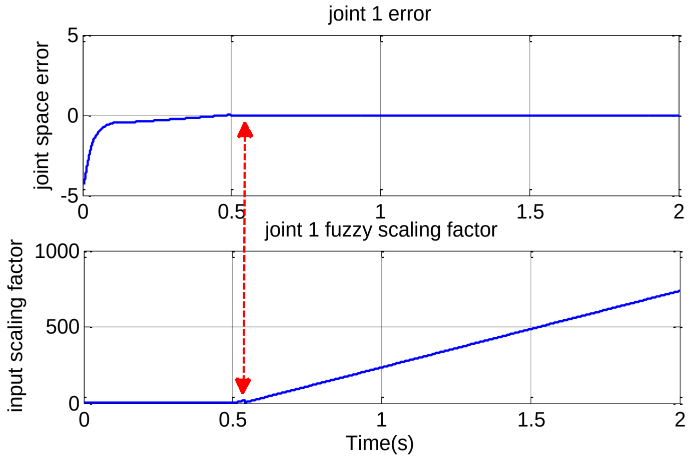

Figure 9. Scaling factor mechanism vs, joint1 error in simulation A

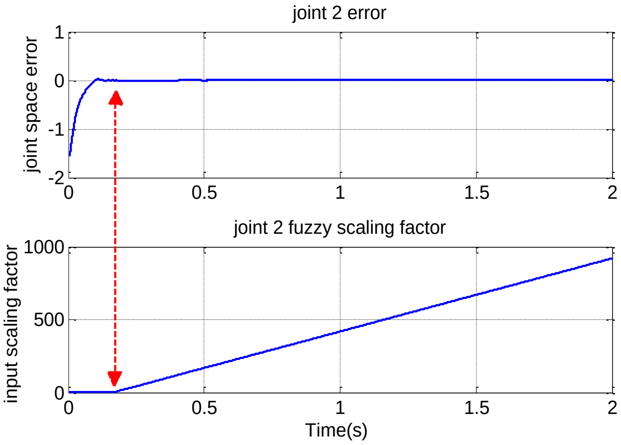

Figure 10. Scaling factor mechanism vs, joint2 error in simulation A

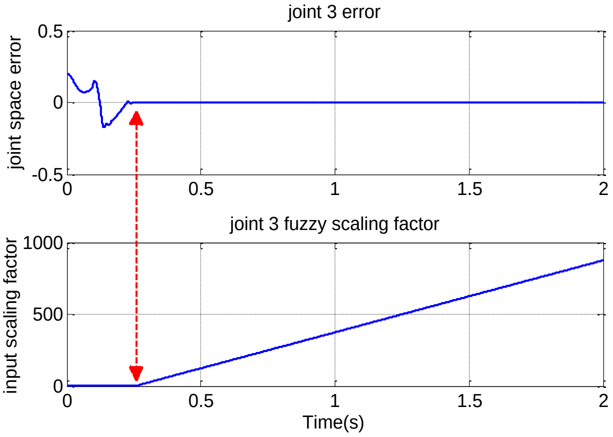

Figure 11. Scaling factor mechanism vs, joint3error in simulation A

An external pulse displayed in Figure 12 is also added to position x as a disturbance in time 1.2 s. The amplitude of pulse is 0.005(m) and its duration time is 0.1s. The disturbance signal represents a sudden movement in direction X. The simulation shows that the fuzzy control with variable scaling factors can perform well against this uncertainty. Assume that the Denavit-Hartenberg parameters used in controller have up to 50 % uncertainty. New parameters are thus introduced as:

$\tilde{a}_2 = 0.5a_2$

$\tilde{a}_3 = 0.8a_3$

$\tilde{d}_3 = 1.4d_3$

where a_2, a_3 , d_3 are exact robot parameters and $\tilde{a}_2$, $\tilde{a}_3$, $\tilde{d}_3$ are parameters that are used for calculating the Jacobian matrix in controller. The controller is applied to the system, and the position of robot end effector is displayed in Figure 13. Although, it takes more time to converge the desired path, it can be observed that trajectory tracking has been performed successfully. A small error is happened in bottom and left side of the circle caused by added external disturbance in time 1.2 s. The tracking error in task space is shown in Figure 14. The fuzzy systems' scaling factors for three joints are displayed in Figure 15. The increasing mechanism has been stopped, and the scaling factors are reset to initial values when the disturbance is added to system in time 1.2 s. Increasing the input scaling factors, intensify the system sensitivity to inputs. So, whenever the tracking error increases, the scaling factors should be reset to their initial values to reduce the sensitivity and to improve the system performance. From all simulations, it can be found that the proposed control system has acceptable performance for tracking a desired path in task space even in existence of kinematic uncertainties.

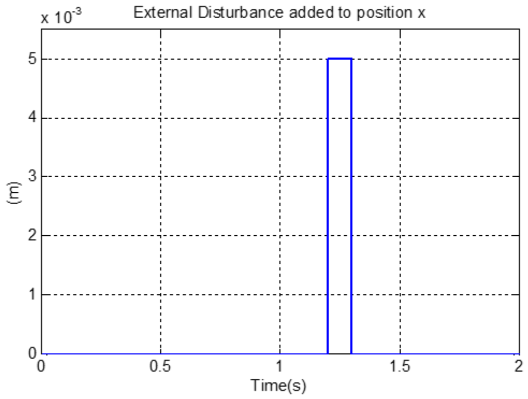

Figure 12. The external disturbance added to position x in simulation B

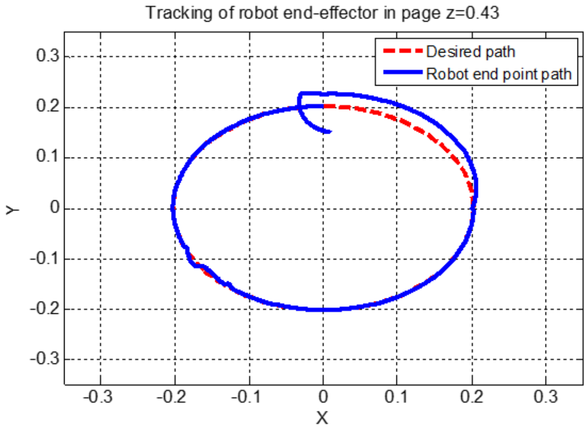

Figure 13. Positon of robot end effector in task space considering up to 50 % uncertainty in parameters and adding an external disturbance (simulation B)

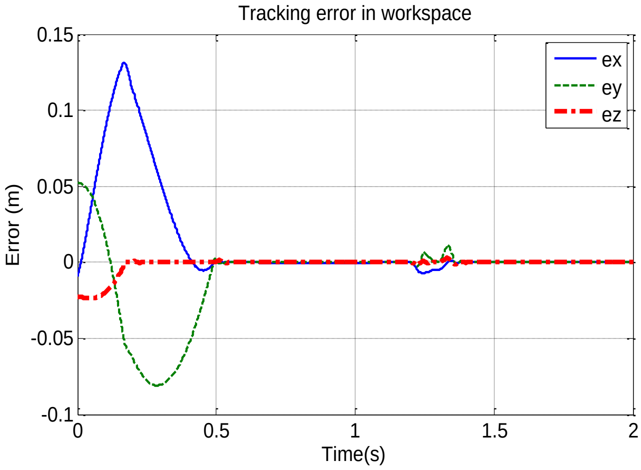

Figure 14. Work space tracking error in simulation B

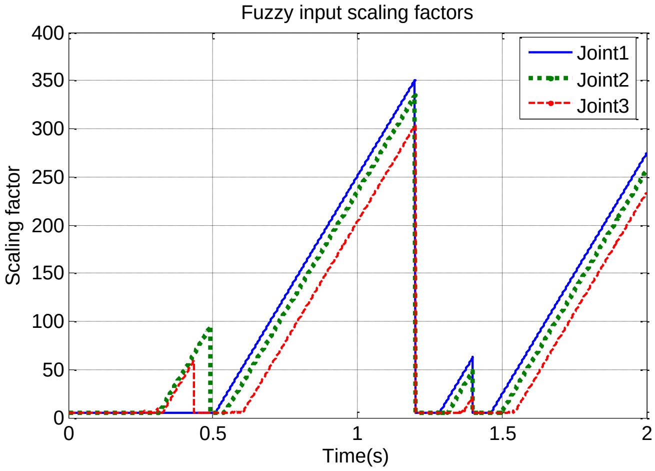

Figure 15. The inputs' scaling factors in simulation B, A disturbance is added to the system in time 1.2s

6. CONCLUSION

A self-tuning fuzzy control system is presented in this paper with the objective to decrease the robot end-effector tracking error. An update mechanism is added to typical fuzzy system in which when the error is small enough the controller inputs' scaling factors are increasing linearly. So there is an adaptive PD controller in zero sets of fuzzy inputs with the ability of decreasing the error. The control system is not dependent to exact model of the system. Only, the feedbacks from task space of robot are needed, and by using inverse Jacobian method the errors are transferred to joint space. The robot end-effector can track a desired trajectory in Cartesian space with uncertain kinematics. It should be noticed that the scaling factor update mechanism is just designed to reduce the tracking error, and overcoming the uncertainties is in-built ability of fuzzy control systems. The simulation results illustrate the acceptable performance of the presented control system.

REFERENCES

[1] S. G. Tzafestas, "9-Mobile Robot Control V: Vision-Based Methods," In Introduction to Mobile Robot Control, ed Oxford: Elsevier, 2014, pp. 319-384.

[2] J. Lin, C. C. Lin, and H. S. Lo, "Pseudo-inverse Jacobian control with grey relational analysis for robot manipulators mounted on oscillatory bases," Journal of Sound and Vibration, vol. 326, pp. 421-437, 10/9/2009.

[3] C.-C. Cheah, C. Liu, and J.-J. E. Slotine, "*Adaptive Jacobian tracking control of robots with uncertainties in kinematic, dynamic and actuator models*," IEEE transactions on automatic control, vol. 51, pp. 1024-1029, 2006.

[4] C.-C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto, "*Approximate Jacobian control for robots with uncertain kinematics and dynamics,*" IEEE transactions on robotics and automation, vol. 19, pp. 692-702, 2003.

[5] C. C. Cheah, S. Kawamura, and S. Arimoto, "*Feedback control for robotic manipulator with an uncertain Jacobian matrix,*" Journal of Robotic Systems, vol. 16, pp. 119-134, 1999.

[6] W. E. Dixon, "*Adaptive regulation of amplitude limited robot manipulators with uncertain kinematics and dynamics,*" IEEE Transactions on Automatic Control, vol. 52, pp. 488-493, 2007.

[7] N. Mendes and P. Neto, "*Indirect adaptive fuzzy control for industrial robots: a solution for contact applications,*" Expert Systems with Applications, vol. 42, pp. 8929-8935, 2015.

[8] I. F. Jasim, P. W. Plapper, and H. Voos, "*Adaptive sliding mode fuzzy control for unknown robots with arbitrarily-switched constraints,*" Mechatronics, vol. 30, pp. 174-186, 2015.

[9] M. Vijay and D. Jena, "PSO based neuro fuzzy sliding mode control for a robot manipulator," Journal of Electrical Systems and Information Technology, 2016.

[10] S. A. Ahmed and M. G. Petrov, "Trajectory Control of Mobile Robots using Type-2 Fuzzy-Neural PID Controller," IFAC-PapersOnLine, vol. 48, pp. 138-143, 2015.

[11] S. Hayati and M. Mirmirani, "Improving the absolute positioning accuracy of robot manipulators," Journal of Robotic Systems, vol. 2, pp. 397-413, 1985.

[12] Y. Jin, "*Decentralized adaptive fuzzy control of robot manipulators,*" IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 28, pp. 47-57, 1998.

[13] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot modeling and control vol. 3: Wiley New York, 2006.

[14] M. M. Fateh and A. Azarfar, "Improving fuzzy control of robot manipulators by increasing scaling factors," ICIC Express Letters, vol. 3, pp. 513-518, 2009.